A2-Central™

formerly Open-Apple

April 1993
Vol. 9, No. 3

A journal and exchange of Apple II discoveries

ISSN 0885-4017
newstand price: $3.00
photocopy charge per page: $0.25

Apple Writer: WPL

by Ron Evry

Up to this point in the guided tour, we have seen how simple, yet versatile, a word processing program *Apple Writer* can be. We have explored how to customize this freeware program with the built-in glossary feature, and seen how virtually every feature of the program is memory-resident; that is, once the program is loaded in, the disk can be removed from a drive and never need to be accessed again. For Apple II owners with one 5.25 disk drive who are used to swapping and flipping disks with other word processors, this is something of a miracle.

But here, at the last stop in the tour, we will examine a "bonus" miracle: WPL, the Word Processing Language that also loads right into your computer's memory along with everything else. Please understand that WPL is not just another menu, or a few extra word processing features. It is an entire programming language that surpasses Applesoft BASIC in its text-handling abilities and is considerably easier to learn.

WPL programs follow a simple structure and contain built-in error checking routines. Since they are written as ordinary text files, the programs can easily be modified and experimented with from within *Apple Writer*. Any WPL program can be chained to others without limit; can call up glossaries, Print/Program Value files and commands; can load, append and delete files or parts of files; and can even be set to load in at bootup on a customized *Apple Writer* disk.

Up until now, WPL has not been a major Apple II programming language, because you needed to buy the *Apple Writer* program (which has been unavailable for years) to use it. Now that *Apple Writer* is freeware, it is possible for every Apple II owner to use WPL programs, even if they have no desire to write them. With that in mind, we will first explore the four very useful WPL programs that come on the *Apple Writer* disk.

To run a WPL program, you will need to have the disk where the program resides in a drive. This is one of those occasions where you will need the *Apple Writer* disk on-line, but it is possible to copy any of these WPL programs onto any directory of any disk you choose.

AUTOPRINT enables you to select up to thirty files from disks and print them at one time. By using this program, you can write a series of documents in one session then print them all at once while you take a coffee break. To use it, follow the standard procedure for running WPL programs. First, press Control-P. Then type "DO" followed by the WPL program name, and finally, press RETURN. You may use either upper or lower case, it does not make any difference. In this instance, your entry may look like this:

```
[P]rint/Program:DO /AW2MASTER/AUTOPRINT
```

If you have previously set the prefix to the directory where the WPL program resides, then you simply have to type "do autoprint" instead of the entire ProDOS pathname. By the way, you do not have to put a space in between "do" and the WPL filename, but you probably should to avoid confusion. Also keep in mind that you should have no unsaved document in memory when you run this program, or it will be lost.

Once you have started AUTOPRINT, you will be prompted for the first filename you wish to print. Here you must type in the full ProDOS pathname, unless, once again, you have previously selected it as a prefix. It is a very good idea to do just that, because AUTOPRINT actually creates a very small second WPL program called PRINTIT and saves it to disk. PRINTIT is only one block long, but space is at a premium on the *Apple Writer* program disk.

Next, you will be asked to enter the next filename, and so on. When you have entered all of the filenames, press RETURN on a blank line and the program will print all of the documents you have entered. If you are writing your programs with the idea of using AUTOPRINT to print them, it is important that you end each of your documents with an embedded form feed (.FF). In fact, it is probably a good habit to get into with all of your documents, no matter how they are printed.

Imagine you have written a series of articles, or chapters for a book, and you have saved them to disk as separate files. Now you want to print them together. CONTPRINT will do this for you, even if the total number of bytes of all of your files adds up to more than *Apple Writer*'s memory limit! In addition, if you have set the Print/Program values to print page numbers, this WPL program will continue numbering from one document to the next.

The procedure for running CONTPRINT is exactly the same as AUTOPRINT's, except you would type "do contprint" at the Control-P prompt. Then just type in your filenames, let your printer loose, and

"GREAT TAX PROGRAM, PHIL!"

go take a break.

The third WPL program on the disk is called COUNTER and it does exactly that. Running it will tell you exactly how many words there are on a document on disk. Once again, save anything you have in memory before running COUNTER, or you will lose it. At the CONTROL-P prompt, type "do counter", type in the name or pathname of the file you need counted, and go away for awhile. This program is **slow,** especially if you have a long file. But it will give you a very accurate count of the words in a file (hint: please do NOT impatiently press RETURN or anything else while waiting for the count; if you do, the count will appear and disappear later on while you are blinking).

The fourth WPL program on the *Apple Writer* disk is called AUTOLETTER. This program enables you to use a prepared list of names, addresses and what-have-you, and insert them into your writing wherever you like, for mass mailings, reports or other personalized documents. To see how AUTOLETTER works, there is a program called DEMOS on the *Apple Writer* disk. Run it by typing "do demos" at the CONTROL-P prompt. The prefix MUST be set for "/AW2MASTER" to run DEMOS, because it accesses a number of programs on that disk. Once you have run it, you can follow the prompts to see a demonstration of AUTOLETTER in action.

To actually use AUTOLETTER or to create your own similar WPL programs, we need to examine how and why the program works. First, we should examine the AUTOLETTER program itself:

```
START   PSX 1
LOOP    NY
        LFORMLETTER
        B
        F/(Address)//
        Y?
        LADDRS!<(X)>!<!N
        PGO FOUND
        PGO QUIT
FOUND   PLSADDRS!<(X)>! !N=$A
        B
        F/(Name)/$A/A
        PNP
        PSX +1
        PGO LOOP
QUIT    PINL  Done at address (X) (press RETURN)
        NY
```

Right off the bat you should notice that there are four words set off from the rest of the program at the far left margin. These four words, START, LOOP, FOUND and QUIT are called LABELS. LABELS can actually be any combination of characters, except for spaces, and they can be any length you wish. Generally, it is good practice to keep LABELS short, to avoid confusion.

At any point in a WPL program, you can jump to any section within it by referring to the LABEL. The command you would use is PGO LABELNAME. In the above program, you will notice the commands PGO FOUND, PGO QUIT and PGO LOOP. PGO commands are case sensitive, so you could actually have LABELS like LOOP, loop and Loop all within the same program.

The rest of the program consists of commands and statements, set off from the LABELS. Any line that is at least one space away from the left margin will be considered a command, but as a rule, most WPL programs have the commands and statements tab-indented for neatness. There is a carriage return at the end of every line.

Looking at the first section, START, we find only one command: PSX 1. This tells *Apple Writer* to set the numeric variable, "X", to a value of "1". With no other commands in START, the program carries out the next instruction, which is the first line of LOOP. Here we see "NY". This does not mean "New York". It is a WPL control command, the equivalent of Control-N ("NEW"), followed by the argument "Y" (for "Yes"). This clears the screen and empties memory for a new document.

The next line, LFORMLETTER, is actually the equivalent of CONTROL-L ("LOAD") followed by the pathname FORMLETTER. It could just as easily been L/HARD1/DOCUMENTS/FORMLETTER, if that was the pathname you needed to load in at this point.

Here you may be noticing a pattern. All control characters can be entered in WPL simply by typing in the letter as a command. If you look at the next line in LOOP, all you will see is the letter "B". This is, of course, the same as a CONTROL-B, which moves the cursor to the beginning of the document in memory and sets the direction arrow on the top bar to the right.

The next command, F/(Address)//, makes sense if you look at the document FORMLETTER:

(Address)

Dear (Name):

Congratulations on your purchase of an Apple computer. You and your family will spend many enjoyable and instructive hours with your new personal computer. In today's fast-paced high-technology world, (Name), you can't afford to be without one. And you can rest assured that when you use an Apple computer, you're using the best there is.

Best wishes,

The Folks at Apple Computer

.inAddress number (X) (press return)

.FF

The "F" is the "FIND" command, which is set to look for the string "(Address)" in the document and replace it with nothing. The next line has a "Y" to automatically answer the prompt, (F)ind:RETURN = PROCEED / Y = REPLACE. Usually, you would press the SPACEBAR to discontinue the (F)ind search (inasmuch as a carriage return will continue it), but any key will work, and WPL will not recognize a space as a command. Therefore, a question mark is substituted here.

The next command, LADDRS!<(X)>!<!N seems like gibberish, but is actually a series of instructions that you may have become familiar with in the preceding parts of this guided tour. The first letter is a (L)oad command, followed by the filename ADDRS and some strings in delimiters which tell it where to load from and where to stop loading. Remember, *Apple Writer* can search any text file on a disk and pull out any part of it you choose, and drop it off wherever your cursor sits on the document. And here we have dropped the cursor off exactly where it took out the string "(Address)".

The exclamation points are used as delimiters here, but there are others you can use if you are searching for strings that happen to have exclamation marks in them. Between the first set of delimiters is "<(X)>", which tells *Apple Writer* to look for a "less than" symbol followed by the numerical value of "X", and the "more than" symbol. The ending marker, between the next two exclamation points, is another "less than" symbol. The letter "N" at the end of the command is a special way of telling *Apple Writer* not to load in the markers, just the text between them. This command can be used in an immediate mode (L)oad command as well; it is not exclusive to WPL.

Looking at the file "ADDRS", we can see how it is laid out to work with this WPL program:

```
<1>John Smith
123 Elm Street
Anytown, U.S.A. 12345
<2>Terry Jones
321 Palm Lane
Centerville, FL 54321
<3>Egbert Q. Manly
1984 Orwell Place
Future, PA 14151
```

```
<4>Harry Q. Public
1953 Warren Court
Sublime, WI 09876
<5>Mary Sanders
0000 Null Result
Meander, OH 54637
<
```

Only the first name and address are loaded from the file on the disk, and inserted into the document where "(Address)" was. The next two commands, PGO FOUND and PGO QUIT are actually linked with each other by the way WPL commands are executed. The first statement tells WPL to go to the section labeled FOUND. But if the program is unable to carry out the previous command for any reason (in this example, we will eventually run out of strings to load), then WPL skips to the next line, PGO QUIT. This is called CONDITIONAL EXECUTION. When WPL comes across an error, it simply considers it an excuse to jump one command line.

Since we are tracking the progress of the AUTOLETTER program, we will jump to the next section, FOUND. Here, we will be attempting to do something different. We will be loading a string ($A) from the text file on disk to use in more than one place in the letter. The first command, PLSADDRS!<(X)>! !N=$A, begins with PLS, the WPL command for "Load String". Here it looks for the beginning marker "<(X)>", as before, but sets the first space as the ending marker. Once again, the "N" says not to load the markers along with the string, and finally the "=$A" sets the value of $A (in this case it would be "John").

You've probably already figured out that the "B" on the next line sends the cursor to the top of the document again, and "F/(Name)/$A/A" replaces "(Name)" with "John" everywhere in the letter (this is starting to get easy, right?). The next line, PNP, is the "New Print" command which sends your document spinning out of your printer all personalized. You may notice the line ".inAddress number (X) (press return)" at the end of the letter. This is the input command, which will stop your printer, flash a message on your screen telling you the address number from your list (X) that you have printed, and then wait for you to press RETURN before continuing. This comes in handy if you are using single sheet feed in your printer. If you use pinfeed sheets, then you may want to not have a line like this in your document. Once you have done that, PSX +1 raises the value of the variable X (this time around to 2), then PGO LOOP sends you back to the section LOOP to start the next letter.

This procedure will continue until you have no more addresses to load in. Then the program will go to the section called QUIT. There, the PIN command works just like the ".in" command on the document. The message "Done at address (X) (press return)" flashes on the screen for you. When you press RETURN, the last command of the program, NY clears memory and then the WPL program ends, having no more instructions.

Understanding how AUTOLETTER works will enable you to create your own customized WPL programs that will insert whatever information you like into letters, contracts, advertisements or anything else. But there is a lot more you can do with WPL. Now that you are familiar with the basic layout of a WPL program, we need to find out what each available command is available for your programs.

First, remember that you can use virtually every *Apple Writer* control character as a WPL command, from A (Adjust screen display) to Z (Toggle word wraparound). Even the dash command (display page and line number) works. Just make it the first letter in a command line, and do not press Control. In addition to this, there are a number of special commands that only are used in WPL.

The WPL-only commands are divided into four groups. The first group consists of the STRING VARIABLE COMMANDS. These include PAS, PLS and PCS. There are only four string variables available in WPL: $A, $B, $C and $D (upper and lowercase are the same). Strings are simply inserted wherever necessary. If the string $A equalled "Ron" and $B stood for "April", then a print statement might look like this:

```
PPRProgrammer of the Month for $B is $A.
```

This would print on the screen as:

```
Programmer of the month for April is Ron.
```

Since string values can be reassigned over and over again, four string variables are usually more than enough for any WPL program. PAS ("assign string") actually gives a value to a string variable. You use PAS like so:

```
PASRon Evry=$A
PAS1st Quarter=$b
PAS/HARD1/LETTERS/MID.TERMS=$c
```

If you put a space in front or in back of the string, the string will include the space. You can concatenate strings (combine two or more together) quite easily:

```
PASMrs. $A=$b
PASMr. $B $C=$D
PASComputer Club of $a=$a
```

No single string can be more than 64 characters total, including the printed out length of concatenated variables.

The command PLS ("load string") enables you to pull a string variable out of a file on disk, as shown in the AUTOLETTER example. You need to use delimiters at the end of the filename to show beginning and ending markers. As in the example, an "N" afterward will only load between the markers, not the markers themselves. Additionally, you can use an "A" after the filename to signify that you want everything loaded that has the same markers, and an "AN" to load everything and leave the markers out. The "A" command is probably more useful in loading an entire file (like a mailing list) with the "L" command than when loading a string, but it is there to use.

If you want to include a carriage return or an exclamation point in your markers, you will need to use alternate delimiters. Neither a slash nor an exclamation mark can utilize the carriage return. If you use "<" as a delimiter, then use ">" as a carriage return character. Others include:

Delimiter	Carriage Return Character
#	%
&	(
*	,

As mentioned before, if a PLS command fails to find the specified string, then the next statement in the program is skipped over. This conditional execution feature makes it possible for you to jump ahead to another routine when you have run out of data.

That brings us to the third string variable command, PCS ("compare strings"), which is always a conditional command. Simply put, this command compares two strings, and if they are not exactly alike in every respect (including case), then the program will skip a line. Delimiters are used to separate the two strings and the first non-space character after PCS is the delimiter. Here are some examples:

```
pcs /$c/Ron Evry/
PCS @$a@$d@
PCS   !Dr. $B!$A!
```

If you have numeric variable to compare with a string variable, or two numerics to compare, you must convert the numeric variable to a string first. We will see how to do this in a bit.

WPL is not a language for extensive mathematical calculations. There are only three numeric variables available at any one time: (X), (Y) and (Z) (not case sensitive). The value of a variable must be an integer between zero and 65,535. While a variable can be set to zero at any time, if its value is increased or decreased to zero by a math function in the program, conditional execution comes into play,

skipping a programming line. Oddly enough, in WPL, 65,535 + 1 =0! The three numeric variable commands are PSX, PSY and PSZ ("set X", "set Y" and "set Z"). If the command is unsigned, it simply sets the variable as that number:

```
PSX 52

PSY 200
```

If the command is signed, it will alter the value of the variable, thus:

```
PSX +8
```

encountered after the above will make the value of (X) sixty, and:

```
PSY -50
```

makes (Y) 150. Please note that if an initial numeric variable was not set before using a signed command, then an error will result and the next line will be skipped. Additionally, you can extract the numeric value of a string (assuming the string is a number) like so:

```
PSZ $A
```

will enable you to perform calculations on the number. To convert a numeric variable to a string variable use the PAS command like so:

```
PAS(Z)=$A
```

By converting numerics to strings, you can compare two numeric variables with the PCS command.

There are four commands unique to WPL that control output to the screen. PPR will print a message on your screen of up to 128 characters. If there is no message after PPR, then a blank line will be printed. Here are some examples:

```
PPR CHOOSE AN OPTION

PPR Ready to print letter to $C.
```

A very important output command is PIN. This is similar to the embedded-in-document command ".IN", except that it is used as part of a WPL program. This command will cause the program to pause until a RETURN is pressed. In certain instances, this command can be used to input a response from the keyboard, enabling the user to type in a variable. Here are some examples:

```
PIN INSERT NEW PAPER IN PRINTER THEN PRESS RETURN
PIN TYPE NAME OF RECIPIENT OF LETTER =$A
PIN ENTER NUMBER OF COPIES =(X)
```

The PIN command works just like PAS or PSX in the above examples. When using a WPL program, you may not want to display your document on the screen. It usually is not necessary, and its presence will slow down execution of the program. In addition, eliminating the text display will give you room to print lots of lines on the screen at once from within the WPL program, enabling you to create on screen menus. In fact, the on screen help menu available with Open Apple-? is actually a WPL program! To eliminate the screen display of the text file, use PND ("no display"). To restore the text display, use the command PYD ("yes display"). In the PYD mode, only one line of text from WPL is displayed at a time.

In addition to the screen output commands, there are a number of printer output commands available. PNP, as we noted in the AUTOLETTER example, is the "new print" command, which prints a single document. The first page of this document is set at whatever value "PN" ("page number") has. PCP ("continue print") will also print a document, but continues page numbering from the previous file. In addition, you may use any of the commands from the Print/Program menu in a WPL program by putting a "P" in front of it:

```
PLM20
```

sets the left margin to 20 and

```
PUT*
```

makes an asterisk the underline token. The only embedded commands that cannot ever be used from within a WPL program are .FF ("formfeed") and .EP ("enable print"). These must be used from within the text.

Commands that control the sequence of instructions are called CONTROL TRANSFER COMMANDS. One we have used already is "DO". This command can be used from within WPL like so:

```
PDO /AW2MASTER/AUTOLETTER
pdo counter
```

You can use this command to chain WPL programs together. The maximum length of a WPL program is 2,048 characters, including the leading spaces before command lines. If there are footnotes in the document in memory, then the footnote buffer takes up half of this space, leaving you with a limit of 1,024 characters for WPL. But while there is a limit to the size of a single WPL program, there is no limit to how many programs you can chain together. And unlike other programming languages, your variables are not lost when you jump from one program to another. Just remember that each program's execution begins at its first line.

We have also explored the PGO statement in AUTOLETTER. This always refers to a labeled section. You can jump around from one section of the program to another with this command. But if you have an often repeated procedure, you may wish to use subroutines. The PSR command ("subroutine") will send you to the subroutine with a given label. Here is an example:

```
PSR PRINT
```

You will wish to be able to return from a subroutine, or else you could have simply used PGO to get there. To return, include the instruction PRT. Be careful to only enter a subroutine section with a subroutine call. If you encounter a PRT without having PSRed your way in, your WPL program will crash (you will get an 'RT' without 'SR' error message in that instance). Subroutines can be nested up to 32 levels deep, which ought to be enough for anybody. A subroutine is not allowed to call itself, so don't do that.

The last control transfer command is PQT ("quit"), which will stop execution of your WPL program wherever it is encountered. You do not need this command if your program ends on the last line. WPL closes up shop if there are no more instruction lines.

WPL is not as prone to crashing as are many other programming languages. Generally, errors make the program skip a line. If this happens inadvertently, crazy things may happen, but you program will probably keep on going like the Energizer Bunny. Nevertheless, there are some errors that bring everything to a crushing halt. When they happen, you will get a message on screen informing you what the problem is.

We have already discovered the 'RT' without 'SR' message. If you nest more than 32 subroutines without the decency to put in a single PRT, your program will halt and the message, "More than 32 'SR'" will show up on the screen. If your program is too long, it will crash right off the bat and you will get the message "Program > 2048 chars". If you send the program to a label that does not exist with a PGO or PSR, then the crash message will say "Label not found —> xxxxx", (with xxxx being the label called for). Remember that label cases must agree.

The only other program crashing error is "Footnote Overflow". This comes up whenever you have more than 1,024 characters of footnote text on a single page. A simple way to print extra-wordy footnotes is to spread them over more than one page.

By making a backup copy of *Apple Writer*, trashing all of the WPL & demo files on the disk and putting your own WPL program you have named STARTUP on the disk, you can create a self-booting WPL program. You can use this to create a startup menu, load glossaries, or send startup programs to a PostScript printer. Since *Apple Writer* is now freeware, there is no reason you can't

make a self-booting disk that will display a letter, article or story and mail the disk to another Apple II owner.

The best way to learn WPL is to write and modify your own programs. Here is a simple program called MULTIPRINT that I have found comes in handy for making more than one copy of the same document:

```
START   PIN Please enter the number of copies to print =$a
        PSX 0
PRINT   PNP
        PSX +1
        PAS(X)=$b
        PCS /$b/$a/
        PGO QUIT
        PGO PRINT
QUIT    PQT
```

Type it in, save it as MULTIPRINT and use it whenever you like. If any of you have some good WPL files, I would love to find out about them. I will be putting together a disk of the best ones I can find soon.

As I promised at the beginning of this tour, here are some sources for additional *Apple Writer* stuff. When writing to these people, please be courteous enough to provide a self-addressed stamped envelope if you expect a reply.

Chester H. Page (1707 Merrifields Dr., Silver Spring, MD 20906) has created a shareware disk that patches up a special Apple IIgs version of *Apple Writer* that will use every bit of available RAM, print high resolution pictures in your documents, and print two columns of text or fancy fonts (including italics) on an Imagewriter.

If you have a Postscript compatible Laser printer, you can do things with *Apple Writer* that will blow your socks off. The authority on the subject is Don Lancaster, 3860 First St, Box 809, Thatcher, AZ 85552. He can be contacted at the Postscript Round Table on GEnie (PSRT), where there are many WPL programs available for downloading, or during (Mountain Time) business hours by voice at the *Apple Writer* users help line (602) 428-4073. Don tells me that he has only one or two copies of the *Apple Writer Cookbook* available (they may be gone by the time you read this). This remarkable work not only has tons of type-in patches and WPL programs for *Apple Writer*, but the chapter explaining how he took apart the program and worked out the source code is an amazing exercise in logic and deduction. His book, *The Incredible Secret Money Machine*, is out in a brand new updated edition and is also worth checking out.

In a preceding installment, I said I would explain how to use the mysterious file "CONTROLV" that's found on the *Apple Writer* disk. When you press Control-V followed by a control character, the control character is embedded in your text rather than taken as a command. The next Control-V turns this feature off, which makes it impossible to embed Control-V itself in a glossary. But embedding a Control-V is a necessary trick if you want your glossary file, in turn, to embed control characters in a document for you. The CONTROLV file consists of a single Control-V character. You can load it into a glossary file as many times as you need it. Once you have it in a document, you can also easily make copies of it using standard *Apple Writer* commands.

Also in an earlier section, I mentioned that the Format command may not work on a GS patched *Apple Writer* program. I have since found out that it will indeed work, but only after setting the prefix to "/AW2MASTER".

If you have an Imagewriter II printer, *Apple Writer* is capable of turning out beautiful full color documents with ease. To do this, first load up the glossary "SPECIAL" into the glossary buffer. Then using CONTROL-G to define new entries, set up seven new commands. Beginning with yellow, type capital "Y" for your command key, followed by CONTROL-V, then the ESCAPE key, another CONTROL-V, capital "K", then still another CONTROL-V, the number "1", and one last CONTROL-V. Then press RETURN, and repeat the procedure for each color. For red, I use a capital "R", and make the number "2". For blue (number "3"), I use the letter "L" because both upper and lowercase "B" are already used in the glossary. Number "4" is orange (letter "O"), "5" is green ("G"), "6" is purple ("P"), and I use "Q" for black (number "0"). Save your amended glossary with the Additional Functions menu, and simply press the Open-Apple-Letter combination at any point in the document you wish to type in a color. *Apple Writer* will not print spaces where your control characters are placed and they can be used in combination with other SPECIAL commands like boldface. If you have a plain black ribbon in the Imagewriter, it will ignore the color commands.

*(This installation concludes our series of articles by Ron Evry on **Apple Writer**. Those of you who have been following the series may have been wondering, Who the heck is this guy?*

*Ron Evry taught himself how to compute on a 1k Timex/Sinclair about five years after they stopped making them. His next bit of programming education was with a Mattel Aquarius (about 1k of ram with color graphics) long after they were discontinued as well. Both computers were free. In 1989, he splurged on a Laser 128 (he decided to go with an Apple II compatible because he had picked up **Apple Writer 2.0** at a flea market for 50 cents), with which he taught himself how to write programs. For a couple of years, his column **"Cheap Computing"** ran regularly in Washington Apple Pi Journal. He is now the Apple II disk librarian for that user group. He has produced programs for Balloons Software and he has created two public domain programs, the **ProDOS Graphic Packer** (available on GEnie), which is a high-res graphic compressor, and **PSL.Dual.DOS**, a method of putting old and new Print Shop graphics on the same side of the same disk for schools that use both programs. Most recently, he has produced **The African American Inventors and Scientists Reading Comprehension Program**, available from Three Dimensional Publishing. He believes in the development of a ninety-nine dollar Apple II compatible computer for schools, especially in an age of textbook rental fees. His motto: "It's not the principle; it's the ten cents.")*

Miscellanea

Softdisk Publishing is looking for a regular contributor for a new monthly AppleWorks GS column. Applicants should have in depth knowledge of the application, as the job requires the creation of a new template and accompanying article each month. If you're an AppleWorksGS guru and could use an additional $200 a month, contact Lee Golden, Softdisk Publishing, 606 Common Street, Shreveport, LA 71101, 318-221-2173. Online addresses include GEnie: Softdisk.Inc, Applelink: Softdisk, AOL: Bryan Zak, internet: softdisk@applelink.apple.com.

If you really know how to program the 65816 in assembly and are into games, give the Illusions Gaming Company a call at 415-435-6999 or fax them your resume at 415-435-5999. This company, based in Turburon, Calif. is looking for a programmer to fill out their development team for a SuperNintendo game. Salary is negotiable, royalties and benefits are available.

GEnie has recently rubbed its magic lamp resulting in a number of desirable effects for clients. Daytime rates have been reduced to only $12.50 per hour. Additionally, in order to help lower charges for those customers who must dial long distance, arrangements have been made with SprintNet (for U.S. customers) and Datapac (for Canadian clients). Surcharges for this service are an additional $2.00 per hour for the former and $6.00 per hour for the latter.

SprintNet access numbers can be found by typing *Phone at any numbered-page prompt on GEnie, while Datapac numbers are avail-

able in the white page directories throughout Canada.

Some U.S. customers can't reach either SprintNet or the GEnie network with a local phone call. In order to accommodate them, GEnie has set up an 800 number. There is a $6.00 per hour surcharge for this service at speeds of 2400 baud or slower. This surcharge applies to all system usage, including the Basic services, but should still save money over calling the nearest connection node by long distance.

Extra good news for 9600 baud customers is that the 800 number for 9600-baud service costs the same as 9600 baud access through GEnie's own network, $18 per hour evenings, nights, and weekends and $24.50 daytime. To access GEnie at up to 2400 baud dial 800-362-1296 from your modem; access for 9600 baud can be reached at 800-847-5260.

There has been much renewed interest in *Apple Writer* by Paul Lutus since its release as freeware due to the extensive efforts of Tim Tobin, GEnie's Apple II head librarian. I found a useful tip in the MiniApp'les user group's newsletter not so long ago that I felt was worth passing on. To import the contents of a disk catalog (or subdirectory) into your document, go to the Options menu of *Apple Writer* and select your default disk. Now select Catalog a disk. Instead of hitting return or entering a name, enter the pound sign (#) and hit return. Hitting a return one last time will take you back to your document and your disk's catalog will be there! If you do this procedure when you already have some text in the document, the catalog information will be inserted wherever you left the cursor.

Due to potential trade name conflicts, the name of InSync Software (3035 E. Topaz Circle, Phoenix, AZ 85028-4423), renowned publishers of ProTERM 3.0, has been changed to InTrec Software, effective March 1, 1993.

When the Spring 1993 Apple Catalog found its way to our office recently I was pleased to note that it listed Apple II manuals along with respective part numbers available for purchase. We get a lot of calls from people who purchased second-hand Apple II's that were missing manuals. Our advice had always been to contact authorized Apple dealers to request they search their service parts list for these manuals. Now ordering them is only an 800 number away. If you are in need of your computer's (or ImageWriter II's) manual, simply call 800-795-1000. Manuals available include the Apple IIc-Plus (#B1132), Apple IIe (#B1136), Apple IIgs (#B1152), *Setting Up Your Apple IIc* (#B1120), *ImageWriter II Owner's Manual* (#B1100), and *A Touch of AppleSoft* (#B1131). The catalog states that all manuals are $15.00.

Lawrence Productions, developer of the award winning McGee series, ventures out to bring computer enjoyment to the older siblings of fans of the captivating toddler. This entry is called *The Lost Tribe*, a strategy game that takes you back to the prehistoric time when a volcano destroys your town and you must lead your homeless tribe to an legendary utopic homeland. Look here next month for a more detailed review. Lawrence Productions can be reached at 1800 South 35th, Galesburg, Mich. 49053, 800-421-4157, 616-665-7075.–edr

SFPD

(No, it's not the San Francisco Police Department, it's a new and recurring column designed to keep you informed of the various shareware, freeware and public domain programs for the Apple II.)

Just after the last issue went to press, Steve Chiang of DreamWorld Software uploaded *DuelTris* to the online services. Here is yet another totally commercial quality game for the Apple IIgs that is practically being given away. *DuelTris* is a one- or two-player game with incredible graphics and sound. And it contains some unique options that make it more than just another *Tetris* clone.

Player Two in this game can be the computer or another person. The speed of the falling tiles can be adjusted from "slowest" to "insane" for each player, which can be a nice feature for players of varying abilities. (Set your kid's speed to insane and yours to slow to give you a chance to win for a change!)

Two of the most unique options of this game are the Inverse and DuelLink mode. Inverse is a special piece function that, when activated, reverses your opponent's control keys. Left is right and right is left, it's maddening and awfully hard to deal with! DuelLink is an area below the playing screen which allows a player to send 2-4 lines back to the other player. When the DuelLink option in enabled from the Menu Bar, it functions automatically when a player fills two or more lines in his/her playing area.

The online version won't save your preferences or high scores, reason enough to send in the measly $15.00 shareware fee. More motivation should be the fact that registered users will receive future versions of the game; and if enough fees are returned, Steve is thinking of adding more features such as joystick and modem support and tournament mode.

Dueltris can be found on the major online services and on the March issue of ***A2-Central on Disk***.–*edr*

Ask (or tell) Uncle DOS

Printer options

I am planning on adding an inkjet printer to my Apple IIgs system. I would appreciate any information you could give me on the relative merits of the Apple Stylewriter, Hewlett Packard DeskWriter, and H.P. Deskjet 500. I am currently leaning towards the StyleWriter, primarily because of price and size. Size is important because I am already using an ImageWriter II, in a print muffler, and a C.Itoh 6510. The ImageWriter is used for most jobs, the C.Itoh is used strictly to print pin-feed checks.

I want to use the inkjet for desktop publishing and for printing scanned graphics. Since I am not going to be using the inkjet for simple correspondence, the ability to print from ProDOS 8 is unimportant. I understand that Vitesse has a StyleWriter driver in beta test now, and I am running System 6.0, so I have Apple's driver already. I am currently using and am very pleased with, Vitesse's *Perfect Image* ImageWriter driver. How good is the Apple StyleWriter driver? How does it compare to the H.P. drivers? Is there any way to take advantage of the extra 60 dpi that the StyleWriter offers? Most of my text work on the inkjet will be with TrueType fonts, and memory on my system is not a problem. Any information on print quality, supply cost, and reliability would be helpful.

Jim Gibson
Indido, Calif.

Hopefully readers experienced with these printers will share their knowledge with us.–edr

FastData fix

Referring to our previous correspondence on the subject of the incompatibility of FastData Pro and System 6.0, I am pleased to advise that I have found a simple solution.

Make a small basic "FASTLAUNCH" file as follows:

```
10 Poke 49151,08 : Rem Make it believe it is ProDOS 1.9
20 PRINT CHR$ (4); "PREFIX/FAST/FASTDATA/": Rem My Path
30 PRINT CHR$ (4); "-F.SYSTEM" : Rem launch FASTDATA
```

My path to F.System and FASTLAUNCH is /FAST/FASTDATA/, so my Prosel-16 Editor information would be: Screen title: FASTDATA, Prefix: /FAST?FASTDATA/, Application Path: ?Prosel.16/), Startup:FASTLAUNCH.

Partial disassembly of the 16k FASTDATA file was a pain, but using the ORCA Disassembler I managed to locate all the text. This would have been much easier if ORCA supported *two* find strings, enabling alternating searches of beginning and end of TEXT, which were always the

same. It was then evident that at the beginning of the program the code verified the ProDOS KVERSION and, if wrong, jumped to the message asking you to request a new version from the author. But when I adjusted the code to look for the ProDOS 2.0.1 KVERSION number, the program froze. I then attempted to see if it was because of calls to unofficial routines that could have been at different locations in ProDOS 1.9 and 2.0.1. But the program crashed while it was checking its integrity, which was verified by just changing one letter in a message under ProDOS 1.9. This must be some kind of protection, as the registered user name appears in the Main Screen. From this discovery it was easy to devise the above solution, much easier than to find and defeat the integrity checking...

Guideon Spirytus
Ville d'Avray, France

Yay Jay

I think we should thank Jay for his article last month. Whether it was a "setup" or not, we need to inform people who think programming is beyond their comprehension, that anyone can have their "15 minutes." The future of Apple II software is going to be shareware/freeware, and as such, anything that helps up and coming programmers is in all of our best interests.

One book I'd like to add to Jay's list is *Programming the 65816, including the 6502, 65c02 and 65802* by David Eyes and Ron Lichty, published by Prentice Hall Press, ISBN 0-89303-789-3. This book is the assembly programmers bible. Although quite out of date (published just before the IIgs was released in 1986), it is the definitive authority on the 65C816. I know a lot of people have tried to get hold of it, and I guess it's out of print by now, but if you pester Ron Lichty or Resource Central, I'm sure they could dig up a stash of them from somewhere!

I'd like to add to Jay's article as well and say that when you're releasing your programs as freeware or shareware, make sure you reply to all the mail you get. If someone bothers to write a letter, it means they cared enough to spend time putting pen to paper and popping it in the post. Why not send them back a disk with your other programs on it? What's a disk worth if it gets your work distributed to more Apple II people? And if you want people to send you mail about a program, make it freeware. I've found that without the obligation to include their hard earned dollars in a letter, many more people take the time to send you their thoughts.

Richard Bennett
West Pymble, Australia

Those who have the Lichty and Eyes book should consider themsleves Lichty, I mean lucky. It has been out of print for years already. Neither Ron Lichty nor Resource Central has been unable to locate any. Try your local library. If they don't have it, ask them to get it for you on interlibrary loan.--edr

About face

I wrote some negative comments about your January issue, so it's only fair that I write to say your February and March issues were all anyone could want. After eight years of using, studying, programming and playing with Apple II computers, it's a pleasure to see there are still lots of things to be learned about them, and that there is still a reliable source of that new information.

I especially enjoyed Jay Jennings' article in the March issue. Mr. Jennings writes in a way that shares his experience with the reader, as opposed to lecturing. The article taught what could be taught but was above all fun to read. He expresses his opinion strongly about brands and sources, while making it clear that it was purely his opinion, not an evaluation. That's the kind of information I like to have and the kind I can respect.

Our user group finally has a cartoonist, so I no longer have to envy you for yours. But I must comment that your are extremely fortunate to have someone who consistently turns out refrigerator-door quality cartoons! How about a profile on the artist?

Your SFPD section is very valuable. One suggestion, though. It may not always be possible to arrange but I much prefer comparative reviews when I can find them. I get very little out of a description of a piece of software in a vacuum. If several programs that do similar things are compared, it becomes much easier to decide which one best suits my needs.

A suggestion for a project that needs to be done— there are hundreds of SFPD desk accessories and INIT files out there. It seems to be accepted without question that it would be impossible to test all possible combinations to make a list of incompatibilities. But with one million Apple IIgs owners around, most combinations are being tested unintentionally. Every Apple II users group either receives centrally or has access through one or more of its members to ***A2-Central***. Suppose the editors of the newsletters of all the Apple II users groups were to collect reports of reproducible incompatibilities and send them to some central place for compilation? How much grief could be saved?

Phil Albro
Cary, N.C.

Beware of the Duo

In the March issue of ***A2-Central***, there is a letter regarding PowerBooks. One thing that was overlooked (and has only recently come to light) is that the only internal modem that works in a PowerBook Duo is Apple's own, and they have not shown any intention of releasing the specs so others can make competitive products. I'm not sure but what Duos require a special memory card (as opposed to the different, special cards for non-Duo PowerBooks). What this amounts to is some hidden expenses in the Duo line – which, as you noted, is only really worthwhile with the purchase of at least the floppy adapter. Trying to do everything through his IIgs will only earn Mr. Desloovere a lot of headaches. On top of that, I'm not sure a Duo actually has a standard AppleTalk port. Food for thought, anyway.

Eric Patterson
Salem, Ore.

The single serial port on PowerBook Duo models supports standard LocalTalk connectors without need for an adapter or dock.--tw

X-10 continental style

I was rather excited by the vision of a computerized home that Art Coughlin painted in his article in the April 1992 issue of ***A2-Central***. In Europe, we have 220-240 volt 50-Hertz cycle power supply, so I was skeptic about buying the X-10 system.

Heath Kit didn't have the item. X-10 System USA had no 220V/50 Hertz units but they gave me the address of the manufacturer in HongKong (Tel 00852-3346848/Fax7642437) who would not sell the unit to individuals, only to vendors. Nevertheless, they gave me the address of their European distributor. (Celtel Ltd UK, P.O. Box 135, Basingstroke, RG25 2HZ, UK, voice (44) 256 64324 or (44) 256 474900, fax (44) 256 818064).

The Controller (CP 290) costs 79.95 Pound Sterling, the appliance modules (AM 566) and Lamp modules (LM 565) are 29.95 each. You have to specify the kinds of plugs you want and they have a X-10 System Technical manual for 9.95. Contollers are delivered with appropriate cables and program for the computer you specify. Versions are available for for the MS-DOS machines, Mac's, C64, and Apple II's.

Gerald Steinbach
Gerlenhofen, Germany

Origami to the people!

I admire the unique design and efficiency of the "origami" envelope order forms sent out by ***A2-Central*** with their great newsletter. ***A2-Central's*** order forms are printed on both sides of a single sheet of paper with glued edges which serve as an order form and envelope in one. The following procedure is an extension of that idea and eliminates the need for tearing off the perforated edges of sprocket feed paper, removing paper and repositioning the printer sprockets to print mailing labels and pasting address labels on the correct envelopes. It also eliminates the need for envelopes and address labels.

I use an Apple IIe but this process will work equally well with any Apple II. Additional requirements are AppleWorks, an ImageWriter II, a few pages of sprocket feed paper, a glue stick (common in any stationery store) and possibly UltraMacros, if you really want to take the fast track.

Begin by creating a new word processing file and name it "Sprocket.Env" or any other name you prefer. Press OPEN-APPLE-O and enter the following printer options:

Top Margin, 3.0	(TM, 3)
Left Margin, 0.0	(LM, 0)
Right Margin, 0.0	(RM, 0)
Centered	(CN)

Escape from the options menu, execute 4

carriage returns and enter the following: *a left arrow, 75 hyphens, a right arrow, and carriage return.*

Press OPEN APPLE-O and enter the following printer option:

Unjustified	(UJ)

Escape from the options menu and enter your own three line return address. End with a carriage return.

Press OPEN APPLE-O and enter the following printer option:

Left Margin 3.0	(LM, 3)

Escape from the options menu and execute 7 carriage returns. Then type in the following numbers:

1
2
3

Execute 8 additional carriage returns.

Press OPEN APPLE-O and enter the following printer options:

Left Margin, 0.0	(LM, 0)
Right Margin, 0.0	(RM, 0)
Centered	(CN)

Escape from the options menu and enter the following: *a left arrow, 75 hyphens, a right arrow, and carriage return.*

Press OPEN APPLE-O and enter the following printer options:

New Page	(NP)
Unjustified	(UJ)
Top Margin, 2.0	(TM, 2)
Left Margin, 1.0	(LM, 1)
Right Margin, 1.0	(RM, 1)
Bottom Margin, 1.0	(BM, 1)

Escape from options menu.

Save this file now to the disk or directory you normally use for correspondence.

If you are now ready to write a letter, rename the file to conform with your own correspondence filing system to prevent accidentally overwriting the newly created template.

This file is set up for no letterhead. If you use different page printing options, they should replace the options following the "New Page" command.

To create your first letter, move up the file to the lines containing the numbers 1, 2, and 3. Use the command OPEN APPLE-Y, to delete those numbers, replacing them with the three line destination address of this letter. Enter OPEN APPLE-9 and begin typing the text of the letter.

Print the letter when it's completed. Remove both pages from the printer in one piece. The safest way to do this is to turn the printer off, manually advance the paper and tear off the pages of the new letter, manually roll the paper back to the original index position and turn the printer on.

Place the still connected envelope and letter pages face down on a flat surface. Apply glue stick sparsely to the back side of both sprocket feed edges of the letter page only. Fold the letter page back on the envelope page and press the edges down. Apply glue stick sparsely to the front side of both sprocket feed edges of the letter page (now facing up). Starting at the letter top, carefully fold the letter and envelope pages in thirds at the fold guide lines printed on the envelope page, and again press the glued sprocket edges together. If you have successfully followed these steps, you now have a letter, folded inside of an addressed envelope, made from a single sheet of paper and ready for a stamp. Set the letter aside and allow to dry.

You no longer need envelopes or labels, and your envelope and paper always match. It is no longer necessary to remove paper from your printer to address envelopes so multiple envelopes and letters can be created in one continuous file and printed in a single operation (don't forget to put a "New Page" command following each completed letter in your file). Mailmerge can also be used with a standard letter and envelope template for group mailings.

And the best part of this exercise is that the person who receives your letter will tear and dispose of those pesky sprocket feed edges.

The only problem I have had with this technique is the time and care that must be used in folding and pasting the final result. However, this extra time is easily offset by the time savings in printer handling, printing labels, applying them to individual envelopes and stuffing the envelopes.

Blain Baird
Pacifica, Calif.

A2-Central™

[redacted]

Publisher: **Tom Weishaar**
Editor: **Ellen Rosenberg**

with help from:
Denise Cameron **Dennis Doms** **Sally Dwyer**
Dean Esmay **Peter Haxton** **Mary Johnson**
Jeff Neuer **Jean Weishaar**

A2-Central,—titled ***Open-Apple*** through January, 1989—has been published monthly since January 1985. World-wide prices (in U.S. dollars; airmail delivery included at no additional charge): $34 for 1 year; $60 for 2 years; $84 for 3 years. All back issues are currently available for $2 each; bound, indexed editions of our first six volumes are $14.95 each. Volumes end with the January issue; an index for the prior volume is included with the February issue.

The full text of each issue of ***A2-Central*** is available on 3.5 disks, along with a selection of the best new public domain and shareware files and programs, for $90 a year (newsletter and disk combined). Single disks are $10.

Please send all correspondence to:

A2-Central
P.O. Box 11250
Overland Park, Kansas 66207 U.S.A.

A2-Central is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also copy ***A2-Central*** for distribution to others. The distribution fee is 25 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. We warrant that most of the information in ***A2-Central*** is useful and correct, although drivel and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may cancel their subscription at any time and receive a full refund of their last subscription payment. The unfilled portion of any paid subscription will be refunded even to satisfied subscribers upon request. OUR LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall our company or our contributors be liable for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017 **GEnie mail: A2-CENTRAL**

Last time in *A2-Central-on-disk:*

Directory: /A2.ON.DISK.9303/

Filename	Blocks	Description
PRODOS	35	New version 4.0.1 for P8 boot disks only
Intro.Mar.93	23	Dean's intro
V9.NO2.MAR.93	117	March issue
General.Stuff	1	
.A2ProLamp.02.93	96	GEnieLamp programmer vers.
.Accudrawl.1Demo	268	Broken-fix next month
.HyperC.Sys.3.0	35	Enhanced HyperC startup
.Magic.File.RTC	10	RealTime Conference
.Intermed.Typing	62	Typing drill/practice
.Video.DataBase	48	DB for videotape collectors
.NoSlotClockPtch	15	Patch for No-Slot Clk
IIgs.Stuff	1	
.Disk.Witch3.1.0	185	Comp. file utl in CDA
.DuelTris.v1.00S	548	2 player Tetris-like game--suburb
.ShadowWrite.1.1	66	Complete word processor in an NDA
.Switch.It.RTC	21	RealTime Conference w/Procyon
.Othello.GS	10	Two player game

Last time in Script-Central

Directory:/ScriptCentrl.11/

Filename	Blocks	Description
Script.Cntrl.11	17	Launch this first
Info	3	Info on SEA's
Biorythm.II	70	The fired up version
ArtWerk2.5	161	Spiders, people, babies, scenic background
fortunCookie2.5	636	Your future in HangTime's stack
theLounge	275	Straight from GEnie's RT
frontEnd.11	937	February/March issue
HyperSonic2.5	111	Lots of sounds
rSounder	87	Appends sounds to your stacks
rSounderDox	13	The docs for rSounder
SetPrefixGS	18	XCMD to set the GS/OS prefix to any pathname
SmartQuotes	37	Convert straight quotes to fancy
ZepHistory	53	Backgrooound of Led Zeplin
ZeppelinTheStak	649	Port of Mac stack